トラブル報告書に特有のフィールド情報を用いたBERTの追加学習

労 瑛瑩　山崎 智弘　伊藤 雅弘
株式会社東芝　研究開発センター
知能化システム研究所　アナリティクスAIラボラトリー
{yingying1.lao, tomohiro2.yamasaki, masahiro20.ito}@toshiba.co.jp

概要

本研究では，インフラ分野のトラブル報告書を用いてBERTの追加学習をするためのフィールド分類タスクを提案する．トラブル報告書に特有のフィールド情報を文脈として活用するので，含まれる文の数が少ないトラブル報告書でも，フィールドが持つ文や単語の潜在的な意味(特徴)を文脈として学習することができる．

文ベクトル表現の分布を可視化するとともに，マスクした単語を推測するタスクや報告書からトラブル表現を抽出するタスクで評価実験を行い，提案手法によって追加学習されたBERTの効果を検証した．

1 はじめに

インフラ事業において，電力・化学プラントなどの運用・保守の作業現場で発生したトラブルを記録している報告書が大量に蓄積されている．我々は，過去の大量のトラブル報告書からトラブルに関係するイベント(「配管に亀裂」「水位が低下」など)の抽出，因果関係の推論といった構造化技術の研究開発に注力している[1]．トラブル報告書に記載された情報を構造化することで，「トラブルが発生した部品」や「効果のあった対処」を早期に把握することが可能になり，保守作業現場の生産性向上に貢献できる．

しかし，トラブル報告書には機器番号や業務用語など一般文書には含まれない用語が数多く含まれている．そのため，日本語 Wikipedia で事前学習されたBERT[2]で文や単語の埋め込み表現を取得しても，系列ラベリングによってトラブルに関係するイベントを抽出する下流タスクで性能を上げにくい課題がある．そこで，トラブル報告書からなるコーパスを用いたBERTの追加学習を通して，インフラ分野に適する文や単語の表現を取得することによって，イベント抽出などの下流タスクでの性能を向上させることを狙う．

トラブル文書_id	日付	タイトル	現象	処置	…
文書1	X年X月X日	○○電源異常	部品Aが壊れた，…	設備Cを取り替えた，…	…
文書2	X年X月X日	××監視不良	部品Bが損傷した，…	設備Dを交換した，…	…
文書3	X年X月X日	△△交換	テキスト1，テキスト2…	テキスト1，テキスト2…	…
…	…	…	…	…	…

図 1　トラブル報告書の事例

トラブル報告書では含まれる文の数が少ないという特徴があるため，従来の学習手法[2]ではBERTの追加学習が困難である．

一方，トラブル報告書は現象や処置などの複数の項目から構成され，各項目にはトラブルに関する詳細情報が文字や記号で記載されている(以下，既定の項目および当該項目に書かれたテキストデータをフィールドと呼ぶ)．図1に示すように，トラブル報告書は横方向に沿った複数のフィールドにトラブルに関する詳細情報が記載されている．縦方向の同じフィールドには類似表現が記載されている．このため，横や縦方向を考慮するフィールドの関係性が利用できると考える．

そこで本研究では，トラブル報告書に特有のフィールド情報を文脈として活用してBERTの追加学習をするため，フィールド分類タスク(Field Classification)を提案する．具体的には，従来のNSP(Next Sentence Prediction)の代わりにフィールド分類を通して，特徴ベクトル空間上の潜在的なフィールドの関係性に従ってテキストデータを分類することにより，クラスごとに同じフィールドの関係性をもつテキストのベクトル表現を近似的に学習する．

提案手法によって追加学習されたBERTの効果を検証するため，文のベクトル表現の可視化，マスク単語の推定タスクとトラブルイベント抽出による評価実験を行う．

2 関連研究

近年，事前学習された言語モデルを用いて，様々

な NLP タスクに応じてファインチューニングすることは主流となっている．特に，BERT は汎用性が高い言語モデルの代表として，文書分類や固有表現抽出などの下流タスクで高いパフォーマンスを達成している． BERT の事前学習では，ラベルなしテキストデータを用いて，MLM と NSP の 2 つの訓練タスクで教師なし学習を行う．MLM では，マスクされたトークンを文脈を考慮して予測するタスクである．穴埋め問題を解くことを通して，単語の埋め込み表現を学習する．NSP では，2 つの入力センテンス A と B が連続するか否かを分類するタスクである．入力センテンスの接続関係を考慮して文の埋め込み表現を学習する．ALBERT [3] は BERT よりも軽量なモデルとして提案されている． ALBERT の事前学習では，NSP の代わりに，SOP(sentence-order prediction)で入力 2 つの文の順序の予測を行っている．文脈に依存しない Word2Vec [4]や GloVe[5]などの従来手法より，事前学習された BERT や ALBERT は文脈を考慮した単語の埋め込み表現を取得可能になる．

Gururangan[6]らは，適用分野や下流タスクなどのデータセットを使って追加で事前学習を行うことを提案し，下流タスクの評価でオリジナルの BERT より精度向上の効果を示した．しかし，インフラ分野のトラブル報告書は，以下の理由から，事前学習された BERT や ALBERT を従来のように追加学習することは困難である．トラブル報告書はフィールドに分けてトラブルに関係する詳細情報を記載しているが，フィールドごとに含まれる文数や 1 文書に含まれる全ての文数がいずれも少ないという特徴がある．これらのトラブル報告書から取得できる学習データは順序ありの文ペアが少なく，順序なしの文ペアが多いため, NSP や SOP でモデルの学習に偏りが発生する恐れがある．それに，ランダムに 2 つの文を取って作成された文ペアには異なる情報が混入して，文脈が大きく変わる可能性が考えられ，MLM で学習された単語の埋め込み表現がずれる恐れがある．

3 提案手法

本節では，トラブル報告書に特有のフィールド情報を文脈として活用する BERT の追加学習タスクを提案する．トラブル報告書において，どのようなトラブルが起こったか(現象)，トラブルを解決するために何をしたか(処置)などのフィールドが存在している(図 1)．従って，既定のフィールドに書かれたテキストには当該フィールド情報を持っていると考えられる．例えば，「現象」フィールドに書かれたテキストには，トラブルの発生状況が詳細に記載されている．これらのテキストは「現象」というフィールド情報を持っている．

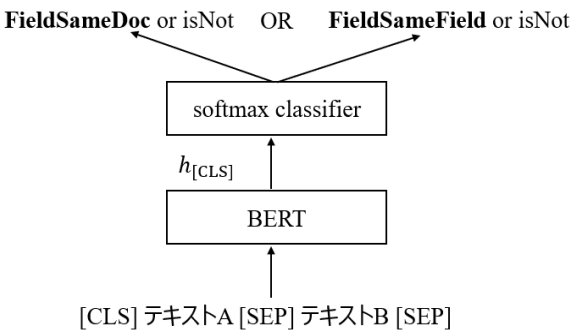


図 2 提案のフィールド分類(Field Classification)

そこで，トラブル報告書における潜在特徴を学習するため，NSP の代わりに，入力テキストのフィールド情報の関係性を学習するフィールド分類タスク(Field Classification)を提案する．図 2 に示すように，提案のフィールド分類は BERT にテキスト A と B を入力して得られた $h_{[CLS]}$を用いて，softmax classifier で 2 文が持つフィールド情報の関係性を分類する．具体的に，フィールドの特性に基づいて入力テキストのフィールド情報の関係性の 2 値分類を行う．フィールド特性とは，入力テキストのフィールド情報の関係性を次の 2 種類で定義する：

フィールドの従属性(fieldSameDoc) 同一文書において，トラブルの発生原因とそれを解決するための対処方法のようにお互いに因果関係を持つ情報が書かれており，異なるフィールドに記載されたテキストでも，潜在的に関係する(図 1 の行に基づく)．従って, 入力のテキスト A と B がそれぞれに属する同一文書に書かれた文か否かというフィールド分類を行い，同一文書の異なるフィールドにある関連情報を潜在特徴として学習する．

フィールドの共通性(fieldSameField) 同一フィールドにおいて，異なる文書で類似するトラブル表現が存在しているため，これらの類似表現は潜在的に関係する(図 1 の列に基づく)．従って，入力のテキスト A と B がそれぞれに属する文書を考慮せず，同じフィールドに書かれた文か否かというフィールド分類を行い，文書をまたがった類似表現の潜在特徴を学習する．

提案のフィールド分類を通して，特徴ベクトル空

間上の潜在的なフィールド特性(fieldSameDoc，fieldSameField のいずれか)に従ってテキストデータを分類することにより，クラスごとに同じフィールド特性をもつテキストのベクトル表現を近似的に学習できる．また，学習のために連続する 2 文をバランスよく取得することが難しいトラブル報告書でも，テキストが属するフィールドの関係性を利用することで 2 文を選択することは容易である．

4 実験と評価

本節では，提案手法の効果を検証するため，実施した実験について述べる．

BERT の追加学習では，電力プラントに関する 3,130 件のトラブル報告書を利用している．具体的に，「タイトル，処置の結果，現象，原因，処置」という 5 種類のフィールドに書かれた文データから，以下 4 種類のデータセットを作成した：

- Dataset1: 属する文書やフィールドの種類を考慮せず，ランダムに 2 文を取得する．計 73k データ数である．
- Dataset2: 同一文書における同一フィールドから連続した 2 文を取得する．計 36k データ数である．
- Dataset3: 属するフィールドの種類を考慮せず，同一文書からランダムに 2 文を取得する．計 73k データ数である．
- Dataset4: 属する文書を考慮せず，同一フィールドからランダムに 2 文を取得する．計 73k データ数である．

東北大学 BERT の bert-base-janpanese-v2[1]を利用して，2 種類のベースライン手法と 2 種類の提案手法によるモデルを追加学習し，性能比較を行う(表 1)．各モデルのパラメータ設定は，最大入力長さを 256，エポック数を 5，学習率を 2×10^{-5}，バッチサイズを 16 とする．なお，学習のエポックごとにモデルを保存する．

4.1 文のベクトル表現の可視化

学習データセットから 50 件のトラブル文書(計 1,147 文) をサンプリングし，追加学習されたモデルでサンプル文のベクトル表現の分布を t-SNE[7]で可視化した．Proposal2 モデルによって取得されたサンプル文のベクトル表現([CLS]トークンであり，768 次元となる)を t-SNE で 2 次元に圧縮した可視化結果を図 3 に示す．図 3 を確認すると，「タイトル」の文のベクトル表現(青)や「処置の結果」の文のベクトル表現(オレンジ)がそれぞれに集まっていて，「現象」(緑)の文のベクトル表現と「処置」(紫)の文のベクトル表現が分離されていることがわかる．

ベースラインの学習手法と比べて，提案するフィールド分類(fieldSameField)を通して，同じフィールドに書かれた文のベクトル表現が互いに近づく，他のフィールドの文のベクトル表現と離れていることがわかる(A 付録参照)．

4.2 マスク単語の推定タスク

提案手法で追加学習されたモデルを用いて，マスクした単語を推定する実験を行った．各モデルの推定結果(A 付録参照)から，ベースラインの学習手法と比べて，提案するフィールド分類(fieldSameDoc, fieldSameField)を通して，特徴ベクトル空間上で同じフィールド特性をもつ文のベクトル表現が互いに近づいていることがわかった．すなわち，これらの文に含まれる単語も近似的に学習されていると考えられる．

4.3 トラブルイベント抽出による評価

提案手法で追加学習されたモデルの性能を評価するため，電力プラントに関するトラブル報告書(日本語)を用いたトラブルイベント抽出の評価実験を行った．トラブルイベント抽出とは，トラブル報告書から「部品 A が壊れた」や「設備 C を取り替えた」のようなトラブルの現象や処置になりうる記述表現をイベントとして抽出するタスクである．

今回利用する報告書には，トラブルに関わるイベントとその因果関係がアノテーションされて，データ数を表 2 に示す．4.1 節で追加学習された BERT モデルを用いて，系列ラベリングによるイベント抽出のモデルを構築した．具体的に，Bi-LSTM-CRF[8, 9]をベースとし，BERT モデルから得られた単語ベクトルを Bi-LSTM への入力として与えた．

表 2 のデータセットを用いてイベント抽出モデルの学習と評価を行い，抽出されたイベント範囲が正解と完全に一致するかどうか(完全一致)，イベント範囲の末尾 5 形態素に一致する(主要部一致)の F 値で評価した．

[1] https://huggingface.co/cl-tohoku/bert-base-japanese-v2

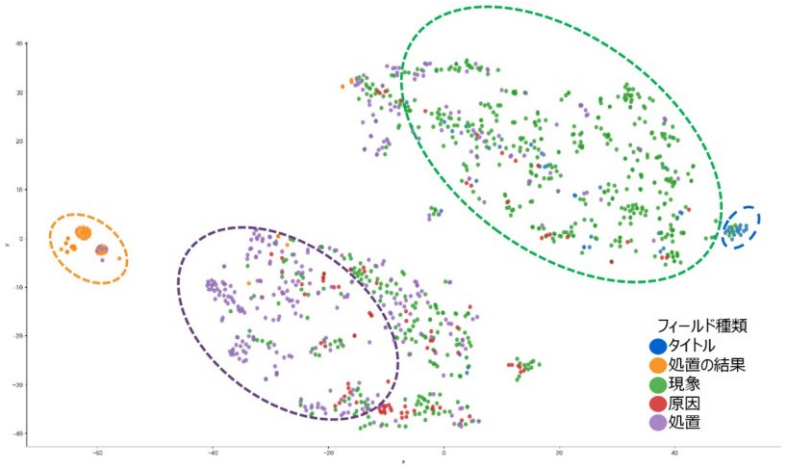


図 3 Proposal2 モデルによるサンプル文のベクトル表現の可視化結果．「タイトル」(青)，「処置の結果」(オレンジ)，「現象」(緑)，「処置」(紫) の点線で集まっている同じフィールドの文のベクトル表現を囲む．

表 1 電力プラントのトラブル報告書を用いた BERT の追加学習の比較実験

比較モデル	学習方法	学習コーパス
Baseline1	MLM only	Dataset1
Baseline2	MLM + NSP	Dataset2
Proposal1	MLM + fieldSameDoc	Dataset3
Proposal2	MLM + fieldSameField	Dataset4

ベースライン手法と提案手法で追加学習されたモデルのエポックごとの完全一致の評価結果を表 3 に示す．表 3 には，Proposal1 モデルは Epoch1~4 で良い性能を示している．特に，Epoch1 の学習結果で Baseline2 モデルより性能評価を 2.3 ポイントで向上した．主要部一致の場合，Proposal2 モデルは Epoch1 と Epoch4 で最も高い評価を得て，Epoch1 の学習結果で Baseline1 モデルより 0.7 ポイントの性能向上を示した(A 付録参照)．

上述の評価結果から，提案タスクでフィールド特性に基づく文や単語の埋め込み表現を近似的に学習することにより，インフラ分野の知識を習得でき，イベント抽出という下流タスクで性能向上に有効性があることが示された．

一方，Baseline1 モデルは，訓練 Epoch 数を増やすにつれて，強い性能を示している．これは，提案手法でフィールド分類 loss と MLM loss の計算スケールが不均衡の恐れで，Epoch 数が増えるほどフィールド分類 loss が小さくなりすぎてモデルの学習への効果が弱くなると考えている．

表 2 イベント抽出用のデータセット

	文書数	文数	イベント数
学習データ	715	7,939	8,887
評価データ	40	515	498

表 3 イベント抽出による評価結果(完全一致)

Epoch	1	2	3	4	5
Baseline1	0.520	0.535	0.539	0.552	**0.576**
Baseline2	0.514	0.526	0.534	0.534	0.559
Proposal1	**0.537**	**0.548**	**0.548**	**0.553**	0.544
Proposal2	0.519	0.531	0.498	0.510	0.514

5 おわりに

本研究では，トラブル報告書に特有のフィールド情報を文脈として活用し，フィールドが持つ文や単語の潜在的な意味(特徴)を学習するフィールド分類タスクを提案した．提案のフィールド分類でインフラ分野のトラブル報告書を用いた BERT の追加学習を行ったところ，イベント抽出という下流タスクでベースラインの学習手法より 2.3 ポイントの性能向上を確認できた．フィールド特性に基づく文や単語の埋め込み表現を近似的に学習することにより，インフラ分野の知識を習得できたと考えられる．今後は，提案するフィールド分類 loss と MLM loss の重み付け計算でモデルの性能を改善し，最適なパラメータ探索を実施する予定である．

参考文献

[1] 伊藤 雅弘，山崎 智弘．アノテーション漏れ推定を用いたエンティティ抽出．言語処理学会 第 27 回年次大会発表論文集(2021 年 3 月). pp1264-1268. 2021.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. Computation and Language, arXiv.1810.04805, 2018.

[3] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, Radu Soricut. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. Computation and Language, arXiv.1909.11942, 2019.

[4] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. Computation and Language, arXiv.1301.3781, 2013.

[5] Jeffrey Pennington, Richard Socher, Christopher Manning. GloVe: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp1532–1543, 2014.

[6] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, Noah A. Smith. Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics(ACL), pp 8342–8360, 2020.

[7] Laurens van der Maaten, Geoffrey Hinton. Visualizing Data using t-SNE. Journal of Machine Learning Research, Vol.9, pp 2579−2605, 2008.

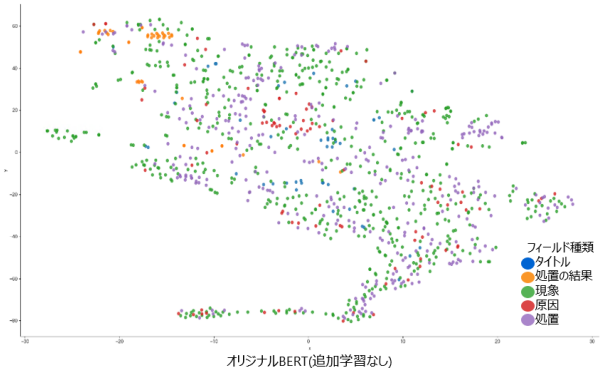
[8] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, Chris Dyer. Neural Architectures for Named Entity Recognition. Proceedings of NAACL 2016, 2016.

[9] Alan Akbik, Duncan Blythe, Roland Vollgraf. Contextual String Embeddings for Sequence Labeling. Proceedings of the 27th International Conference on Computational Linguistics, pp1638–1649, 2018.

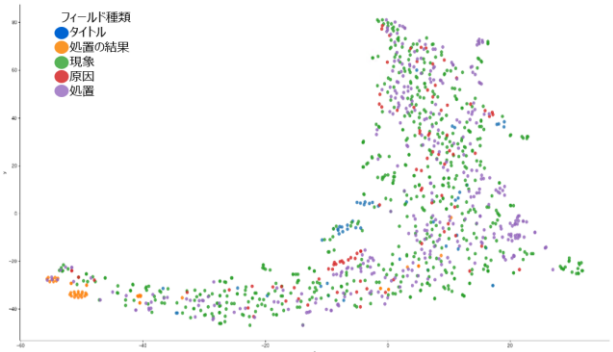
A 付録

文のベクトル表現の可視化

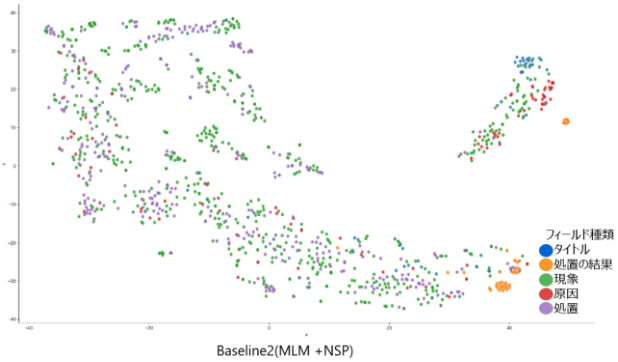
フィールド特性が共通性の場合，50 件のサンプル文書(計 1,147 文) をランダムに取得して，オリジナル BERT(追加学習なし)とベースライン手法で追加学習されたモデルの可視化結果を以下のように示す．



オリジナルBERT(追加学習なし)



Baseline1(MLM only)



Baseline2(MLM +NSP)

マスク単語の推定タスク

提案手法で追加学習されたモデルを用いて，マスクした単語を推定するタスクの結果を考察する．以下の表 4 に，考察事例および各モデルの推定結果(top10)を示す．「現象」に書かれた例文 1 において，Proposal1 モデルは正解である‘故障’と類似する意味を持つ，‘異常’が推定されており，他に‘故障’と含む文書における頻出単語である‘対応’，‘基板’，‘客’，‘常時’，‘CPU’も推定されている．「処置」に書かれた例文 2 において，Proposal2 モデルは正解である‘交換’が推定されており，他に「処置」における頻出単語である‘基板’，‘客(##先)’，‘様子’，‘見’も推定されている．

表 4 マスク単語の推定事例

「現象」例文 1 ：[MASK]推定部位は設備 A である Masked Token 正解：‘故障’	
Baseline1	['CPU', '→', '推定', '基板', '対応', '・', '客', '常時', '表示', '3']
Baseline2	['の', '(', ')', '・', 'て', '-', '[PAD]', '、', '不良', 'に']
Proposal1	['対応', '常時', '→', '異常', '客', '様子', 'CPU', '1', '基板', '・']
Proposal2	['基板', '・', '-', 'の', '1', '不良', ':', '2', '/', 'に']
「処置」例文 2 ：基板[MASK]にて処置済み Masked Token 正解：‘交換’	
Baseline1	['基板', '不良', '##U', ':', 'は', '見', '表示', '部位', '監視', '交換']
Baseline2	[')', '基板', 'の', '不良', ':', '(', '、', '-', 'に', 'た']
Proposal1	['基板', '/', ':', '不良', '##U', '見', '監視', 'CPU', '交換', '##先']
Proposal2	['基板', 'は', 'の', 'を', '様子', '交換', '客', 'に', '見', 'が']

トラブルイベント抽出による評価

抽出されたイベント範囲が正解イベント範囲の末尾 5 形態素に一致する(主要部一致)の F 値で評価した結果である．

表 5 イベント抽出による評価結果 (主要部一致)

Epoch	1	2	3	4	5
Baseline1	0.785	0.791	**0.804**	0.786	**0.803**
Baseline2	0.763	0.782	0.796	0.784	0.787
Proposal1	0.791	**0.792**	0.785	0.782	0.798
Proposal2	**0.792**	0.790	0.780	**0.790**	0.793